

Detecting Firefly Algorithm for Numerical Optimization

Yuchen Zhang¹, Xiujuan Lei^{1(✉)}, and Ying Tan²

¹ School of Computer Science, Shaanxi Normal University,
Xi'an 710119, China

xjlei168@163.com

² School of Electronics Engineering and Computer Science,
Peking University, Beijing 100871, China

Abstract. Firefly Algorithm (FA) is a stochastic optimization algorithm inspired by the swarm intelligence. It has the advantages of simple implementation, high efficiency and so on. However, the algorithm is easy to come into premature convergence and fall into local optimum. To address this problem, we proposed a novel firefly algorithm, Detecting Firefly Algorithm (DFA), in which we use a detecting firefly that flies round certain target points to improve the search path of standards FA. Moreover, the influence of the brightest firefly and the second brightest firefly is taken into consideration to optimize the movement strategy of the single firefly. The example illustrates that the higher precision and better convergence features of the proposed algorithm in numerical optimization.

Keywords: Firefly algorithm · Detecting firefly · Global optima

1 Introduction

Firefly algorithm (FA) is a stochastic optimization algorithm inspired by the swarm intelligence introduced by XinShe Yang [1] at Cambridge University. The algorithm is of higher convergence rate especially in function optimization. A lot of numerical examples have illustrated the superiority of this method. FA has been applied to many different fields. Nggam J. Cheung *et al.* analyzed of the relevant parameters of the algorithm [2]. Xinshe Yang *et al.* also made a further research on the algorithm that some unique test functions were introduced [3]. And the advanced nature of the algorithm was described in the optimization problem [4]. Xinshe Yang believed that it was more effective in solving NP-hard problems [5]. A new method using FA for hybrid data clustering was explained by Maheshwar *et al.* [6]. Shaik Farook used hybrid genetic-firefly algorithm for regulating LFC regulations in a deregulated power system [7].

However, standard FA is easy to fall into the local optimal solution in the global search. In order to overcome this defect of the algorithm, Arora *et al.* proposed mutated firefly algorithm to improve the convergence rate and accurateness [8]. Shuhao Yu *et al.* used a variable strategy for step size α to modify firefly algorithm (VSSFA) [9]. A new disturbance model of the brightest firefly was improved in the paper of S.L. Tilahun *et al.* [10]. And a peculiar movement strategy was described by adding inertia

weight in literature [11]. At the same time, Mahdi Bidar *et al.* described a new method called JFA which promoted the low brightness firefly [12]. The intelligent firefly algorithm (IFA) was introduced by Seif-Eddeen K. Fateen *et al.* in which a firefly acted intelligently by basing its move on top ranking fireflies [13]. Milan Tuba *et al.* also upgraded FA for portfolio optimization problem [14]. In addition, Le'vy Flights was combined with search strategy of FA in literature of XinShe Yang *et al.* [15]. Some scholars also introduced the chaos mechanism into the algorithm [16, 17]. As well, a kind of adaptive firefly optimization algorithm based on stochastic inertia weight was discussed by Changnian Liu [18].

In the above improved methods, the search path of the firefly was not mentioned. In standard FA, a firefly in the search process is not clear about the direction of movement, so that it is not conducive to find the global optimal solution. Therefore, we presented a kind of detecting firefly which surrounds a target point to do a spiral motion, and gradually converges to the target point. When the scope of the definition domain is larger, the more dispersed fireflies will lead to lower convergence rate. In order to solve this problem, we add the item of brightness to the definition of attractiveness. And the influence of the global optimum and the suboptimum also was considered to join in the process of the movement of the firefly.

The rest of the paper is structured as follows. The second section is a review of the standard FA. DFA is described in third section. In fourth section, the proposed algorithm is tested on 28 benchmark functions. Moreover, the simulation result is compared with the standard FA and several intelligent optimization algorithms and then the performance of DFA is verified. The last section summarizes the paper.

2 Standard Firefly Algorithm

For simplicity in describing FA which was developed by XinShe Yang, There are three idealized assumptions as following [1]:

- Each firefly will be attracted to other fireflies only depend on their brightness regardless of their sex or other.
- Any two flashing fireflies, the less bright one will be attracted and move towards the brighter one. The attractiveness of fireflies both decrease as their distance increases. If there is no brighter one than a particular firefly, it will move randomly.
- The brightness of a firefly is affected or determined by the value of the objective function.

In standard FA, the brightness of a firefly is proportional to the value of objective function. The absolute brightness of a firefly at a particular location x is chosen as:

$$I(x) \propto f(x) \tag{1}$$

where $I(x)$ is absolute brightness when the attenuation of brightness in the medium is not considered. As attractiveness of a firefly is inverse proportional to the distance seen by adjacent fireflies, we define the attractiveness β of a firefly by:

$$\beta = \beta_0 e^{-\gamma r^2} \quad (2)$$

where β_0 is the attractiveness at $r = 0$, γ is a light absorption coefficient. The movement strategy of a firefly i is attracted to another more attractive (or brighter) firefly j is determined by:

$$x_i = x_i + \beta(x_j - x_i) + \alpha \varepsilon_i \quad (3)$$

where the second term is due to the attraction. The third term is randomized with α being the randomization parameter, and ε_i is a vector of random numbers drawn from a Gaussian distribution or uniform distribution.

3 Detecting Firefly Algorithm

3.1 Detecting Firefly

Inspired by detecting particle swarm optimization that was proposed by Y Zhang [19], we introduced the detecting firefly to FA. In standard FA, the search path of single firefly is disorganized. It is easy to make the algorithm fall into local optimal solution. The aim of this research is to help the firefly escape from local optimal trap. We selected a small portion of the n fireflies whose number was marked as n_s . These fireflies which are known as detecting firefly try to find a better potential optimal value along the spiral convergence line path (a spiral center is required in the domain of definition). Using the detecting firefly can expand the search scope of the algorithm. In the meantime, the rest of fireflies search the optimal value in accordance with the standard FA. That natural firefly and detecting firefly complement each other effectively to maintain population diversity and avoid premature phenomenon in the evolution.

In order to make the detecting firefly move around the target point (such as current global optimal value) along the path of the spiral line, we used an offset ΔX which is equal to the speed of particle in PSO. When detecting firefly i is going to move, its location update formula is defined as follows:

$$\Delta X_i^{k+1} = C_{s1} \Delta X_i^k + C_{s2} (X_{gbest} - X_i^k) \quad (4)$$

$$X_i^{k+1} = X_i^k + \Delta X_i^{k+1} \quad (5)$$

where X_{gbest} is the position of the brightest firefly, coefficient k represents the k_{th} iteration, C_{s1} and C_{s2} respectively represent the proportion of ΔX_i^k and $(X_{gbest} - X_i^k)$ in movement direction of the detecting firefly and the initial position X_i^0 and initial offset ΔX_i^0 are random. We selected the position of current optimal firefly as the target. n_s ($n_s = 5$) detecting fireflies travel in the spiral track which were given in Fig. 1 around the position of current optimal firefly. This process will be iterated for T ($T = 10$) times. For example, we selected firefly i as a detecting firefly. The original brightness I_i of firefly i and the brightness I_i^* of detecting firefly i^* which is the brightest in T iterations

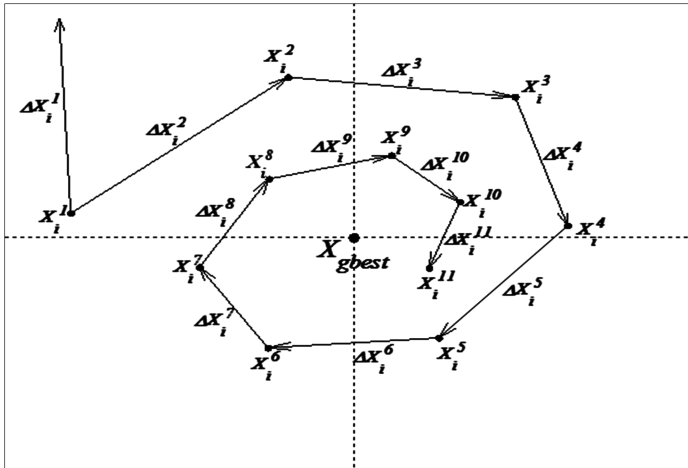


Fig. 1. The search path of the detecting firefly

are compared. If the brightness I_i^* is greater than the original brightness I_i , the algorithm will use the position of detecting firefly i^* to update the original position of firefly i .

3.2 Randomization Parameter (Step Length) and Attractiveness

In fact, the randomization parameter α of the standard FA is fixed in the formula of movement of a firefly. It cannot be perfectly suited to the search process of firefly. When α is large, the algorithm has a satisfied convergence rate. But it is easy to ignore the potential of the optimal value, so that it reduces the accuracy in late search. On the other hand, when α is smaller, the algorithm has the high precision in local. However, reduced search space can result in that the algorithm cannot find the global optimal value. In order to balance the search range and precision, we used an adaptive strategy to generate α [1] as follow:

$$\alpha = \alpha_e + (\alpha_s - \alpha_e)e^{-t} \tag{6}$$

where α_s is initial value, α_e is the final value specified, coefficient t represents the t_{th} iteration. The formula shows that α decreases exponentially with the increase of the iteration.

In standard FA, the movement direction of the firefly is determined by the brightness of each firefly. But the degree of the attractiveness between the fireflies is only related to the distance. It does not reflect the impact of the brightness in the movement of the firefly [9]. If the function optimization problem has a larger scope of definition, a brighter firefly which is far from the others will not have enough attractiveness to the others. So in DFA, a new method of calculating attractiveness was designed depending on both brightness and direction. If firefly j is brighter than firefly i , the attractiveness β_{ij} of firefly j to firefly i is given by:

$$\beta_{ij} = \begin{cases} \frac{I_j}{I_i} e^{-\gamma r^2} & I_i > 0 \\ \beta_o & I_i = 0 \end{cases} \tag{7}$$

where I_i and I_j are absolute brightness of firefly i and firefly j respectively. In this paper, it is important to note that the brightness of all fireflies is greater than zero which depends on the objective function. Therefore, the minimum brightness is zero.

3.3 Firefly Movement

In the position update process of original firefly, the current brightest firefly has not been concerned in the algorithm. XinShe Yang introduced to add an extra item about the brightest firefly into Eq. 3 [1]. But this strategy is likely to cause that the movement direction of the firefly becomes single.

Table 1. The pseudo code of DFA

Detecting Firefly Algorithm:

```

Objective function  $F(X)$ ,  $X = (x_1, \dots, x_d)^T$ 
Generate initial positions of all fireflies  $X_i (i = 1, \dots, n)$ 
Brightness  $I_i$  at  $X_i$  is determined by  $F(X_i)$ 
Define light absorption coefficient  $\square$  number of detecting firefly  $n_s$ 
while ( $t < \text{MaxGeneration}$ )
  Randomly select  $n_s$  fireflies as detecting fireflies
  for  $i = 1:n$  all  $n$  fireflies
    if (firefly  $i$  is detecting firefly)
      Randomly generate initial position of detecting firefly  $i$ 
      for  $k = 1:T$ 
        Update position detecting firefly  $i$  via Eq.4-5
        Evaluate new solutions
      end for  $k$ 
      Determine the brightest firefly  $i^*$  in  $T$  iterations
      if ( $I_i^* > I_i$ )
        Firefly  $i$  is replaced by detecting firefly  $i^*$ 
      end if
    else
      for  $j = 1:n$  all  $n$  fireflies
        if ( $I_j > I_i$ )
          Attractiveness varies with brightness and distance via Eq.7
          Move firefly  $i$  towards  $j$  in  $d$ -dimension according to Eq.8
        end if
        Evaluate new solutions and update brightness
      end for  $j$ 
    end if
  end for  $i$ 
  Rank the fireflies and find the current optimum and the suboptimum
end while
Post process result and visualization

```

DFA was continued to add another item about second brightest in our improvement. When the firefly is moving, the space between the optimum and the suboptimum will be searched. It avoids the premature convergence that is caused by adding the optimal location only. Suppose that the brightness of firefly i is not as good as firefly j , firefly i is attracted by firefly j . The updated location of firefly i is defined by:

$$x_i = x_i + \beta_{ij}(x_j - x_i) + \alpha\varepsilon_{i1} + \lambda_1\varepsilon_{i2}(x_{gbest} - x_i) + \lambda_2\varepsilon_{i3}(x'_{gbest} - x_i) \quad (8)$$

where x_{gbest} is the location of the brightest firefly and x'_{gbest} is the location of the second brightest firefly. $\varepsilon_{i1}, \varepsilon_{i2}$ are vectors of random numbers in $[-0.5, 0.5]$. λ_1, λ_2 are inertia weight. If a firefly is the brightest firefly, it will move randomly. We used the best firefly after disturbing to replace the original firefly. According to Eqs. 4-8, the pseudo code of DFA is given by Table 1.

4 Simulation and Experiments

The algorithm was compared with other intelligent optimization algorithms, such as standard particle swarm algorithm, genetic algorithm, SFA, and VSSFA [9]. The parameters of these algorithm were listed by Table 2. The proposed DFA was tested by 28 benchmark functions in CEC 2013 which be summarized in paper [20]. There are five unimodal functions (from f_1 to f_5), 20 basic multimodal functions (from f_6 to f_{20}) and 8 composition functions (from f_{21} to f_{28}). All problems are the minimum optimization. And the dimensions of all test functions were 30. The scope of each dimension was in $[-100, 100]$. We adopted 50 independent runs for each of test function and then calculated the average of 50 results.

Table 2. The parameters' setting

Parameter	Variable	GA	SPSO	FA	VSSFA	DFA
Cross coefficient	q_c	0.9	-	-	-	-
Mutation coefficient	q_m	0.1	-	-	-	-
Learning factor	c_1	-	2	-	-	-
Learning factor	c_2	-	2	-	-	-
Maximum velocity	v_{max}	-	0.5	-	-	-
weight	w	-	1	-	-	-
Step length	α_0	-	-	0.2	-	-
Initial step length	α_s	-	-	-	-	1
Final step length	α_e	-	-	-	-	0.1
Maximal attractiveness	β_0	-	-	1	1	1
Light absorption coefficient	γ	-	-	1	1	1
Offset weight	C_{s1}	-	-	-	-	0.8
Target weight	C_{s2}	-	-	-	-	0.8
Optimal factor	λ_1	-	-	-	-	0.6
Sub optimal factor	λ_2	-	-	-	-	0.6

The average values on the optimization were presented in Table 3, where the top of the ranked value was highlighted in boldface. From the results, the average ranking of DFA is 1.46 that is far higher than other algorithms in ranking.

Table 3. Simulation result of the 28 benchmark functions

<i>f</i>	Function Average Value and Rank									
	SPSO	Rank	GA	Rank	SFA	Rank	VSSFA	Rank	DFA	Rank
<i>f</i> ₁	-1.400E + 03	1	1.234E + 04	5	-1.392E + 03	3	-1.389E + 03	4	-1.400E + 03	1
<i>f</i> ₂	1.249E + 06	1	2.683E + 08	5	7.971E + 06	3	7.105E + 06	2	8.433E + 06	4
<i>f</i> ₃	6.361E + 08	2	7.164E + 10	5	1.835E + 09	4	9.300E + 08	3	5.088E + 08	1
<i>f</i> ₄	1.615E + 04	1	6.880E + 04	5	2.697E + 04	2	3.534E + 04	3	3.944E + 04	4
<i>f</i> ₅	-9.999E + 02	2	1.018E + 03	5	-9.972E + 02	3	-9.964E + 02	4	-1.000E + 03	1
<i>f</i> ₆	-8.581E + 02	1	3.138E + 02	5	-8.421E + 02	3	-8.368E + 02	4	-8.454E + 02	2
<i>f</i> ₇	3.070E + 05	5	-5.373E + 02	4	-6.615E + 02	3	-7.074E + 02	2	-7.215E + 02	1
<i>f</i> ₈	-6.790E + 02	3	-6.791E + 02	1	-6.786E + 02	5	-6.786E + 02	4	-6.790E + 02	2
<i>f</i> ₉	-5.602E + 02	5	-5.609E + 02	4	-5.718E + 02	3	-5.741E + 02	2	-5.754E + 02	1
<i>f</i> ₁₀	-4.989E + 02	1	1.725E + 03	5	-4.943E + 02	3	-4.953E + 02	4	-4.978E + 02	2
<i>f</i> ₁₁	6.374E + 01	5	-2.301E + 01	4	-3.487E + 02	3	-3.490E + 02	2	-3.609E + 02	1
<i>f</i> ₁₂	1.598E + 02	5	1.463E + 02	4	-1.128E + 02	2	-9.948E + 01	3	-2.109E + 02	1
<i>f</i> ₁₃	3.683E + 02	5	2.632E + 02	4	8.368E + 01	3	5.537E + 01	2	-4.610E + 01	1
<i>f</i> ₁₄	5.604E + 03	4	6.241E + 03	5	1.477E + 03	2	1.563E + 03	3	1.186E + 03	1
<i>f</i> ₁₅	6.363E + 03	4	7.475E + 03	5	4.554E + 03	2	4.512E + 03	1	5.455E + 03	3
<i>f</i> ₁₆	2.026E + 02	2	2.025E + 02	2	2.058E + 02	4	2.060E + 02	5	2.016E + 02	1
<i>f</i> ₁₇	7.763E + 02	4	1.162E + 03	5	5.610E + 02	3	5.475E + 02	2	4.447E + 02	1
<i>f</i> ₁₈	8.940E + 02	4	1.264E + 03	5	6.900E + 02	2	6.927E + 02	3	5.748E + 02	1
<i>f</i> ₁₉	5.526E + 02	4	1.038E + 04	5	5.170E + 02	3	5.163E + 02	2	5.103E + 02	1
<i>f</i> ₂₀	6.149E + 02	4	6.149E + 02	5	6.148E + 02	3	6.145E + 02	2	6.141E + 02	1
<i>f</i> ₂₁	1.043E + 03	3	3.114E + 03	5	1.038E + 03	2	1.059E + 03	4	9.906E + 02	1
<i>f</i> ₂₂	6.439E + 03	4	7.542E + 03	5	2.311E + 03	2	2.427E + 03	3	1.949E + 03	1
<i>f</i> ₂₃	7.584E + 03	4	8.883E + 03	5	6.269E + 03	2	6.114E + 03	1	6.607E + 03	3
<i>f</i> ₂₄	1.398E + 03	5	1.322E + 03	4	1.278E + 03	3	1.275E + 03	2	1.266E + 03	1
<i>f</i> ₂₅	1.509E + 03	5	1.440E + 03	4	1.400E + 03	3	1.397E + 03	2	1.381E + 03	1
<i>f</i> ₂₆	1.550E + 03	5	1.426E + 03	4	1.508E + 03	2	1.531E + 03	3	1.407E + 03	1
<i>f</i> ₂₇	2.808E + 03	5	2.685E + 03	4	2.336E + 03	3	2.263E + 03	2	2.242E + 03	1
<i>f</i> ₂₈	5.453E + 03	5	4.508E + 03	4	2.199E + 03	3	2.139E + 03	2	1.806E + 03	1
Mean	SPSO	3.57	GA	4.39	FA	2.82	VSSFA	2.71	DFA	1.46

Though the performance of DFA is not as good as SPSO on function *f*₂, *f*₄, *f*₆, *f*₁₀, and is worse than GA and VSSFA respectively on function *f*₈ and *f*₁₅, *f*₂₃, the algorithm has an absolute advantage on the rest of test functions. The simulation and experiments show that the performance of the algorithm is significant which beyond other in finding the global optimal value.

To amplify the convergence property of DFA, the convergence curves were displayed at Figs. 2, 3, 4 and 5. Due to space limitations, our paper only described four convergence curves on functions *f*₁, *f*₁₁, *f*₁₃, *f*₂₈. Figures 3 and 4 show that convergence rate of DFA was greater than other algorithms on Sphere function and Rastrigin’s function. On Sphere function, DFA began to converge at the 400_{th} iteration. But SFA and VSSFA still fluctuated at the 800_{th} iteration. DFA began to converge at the 300_{th}

iteration on Rastrigin's function. However, SFA and VSSFA were not convergent in 1000_{th} iterations. On non-continuous rotated Rastrigin's function, DFA is better than FA, not as good as VSSFA. And on the composition function $8(n = 5, \text{rotated})$, the convergence rate of DFA is as high as that of SFA, which is higher than VSSFA.

The experiments displayed that DFA has well accuracy and convergence rate on different function optimization problems. Adding the detecting firefly made the algorithm escape from the local optimal trap and premature convergence. The decline of convergence rate that detecting firefly caused was blocked by adding optimum and suboptimum.

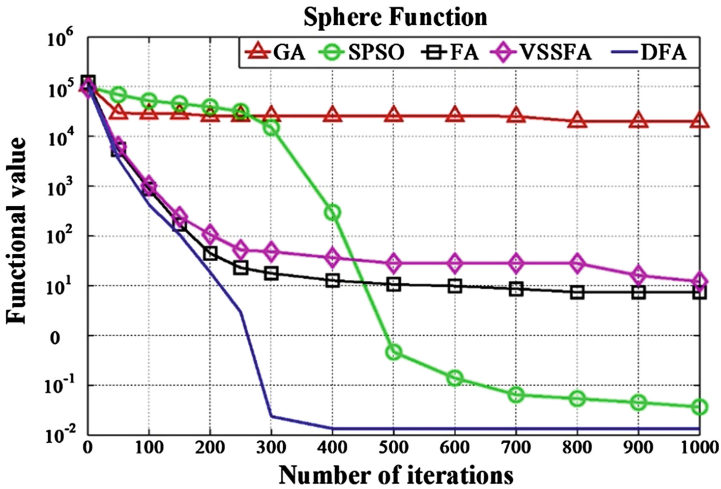


Fig. 2. Convergence curve of f_1

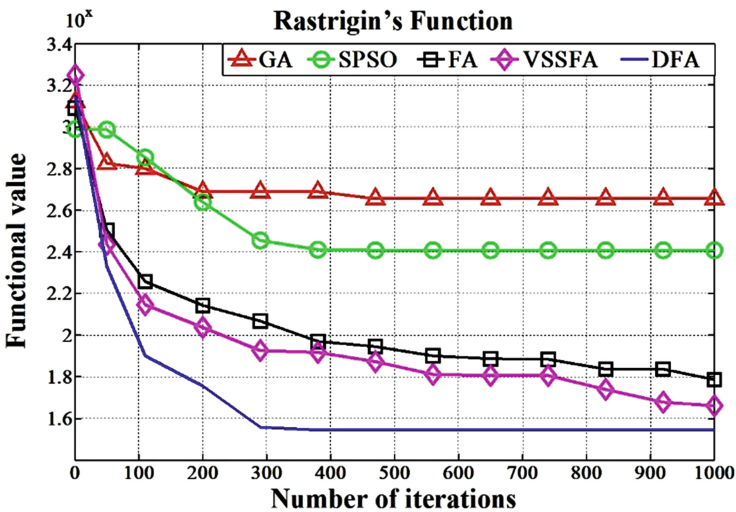


Fig. 3. Convergence curve of f_{11}

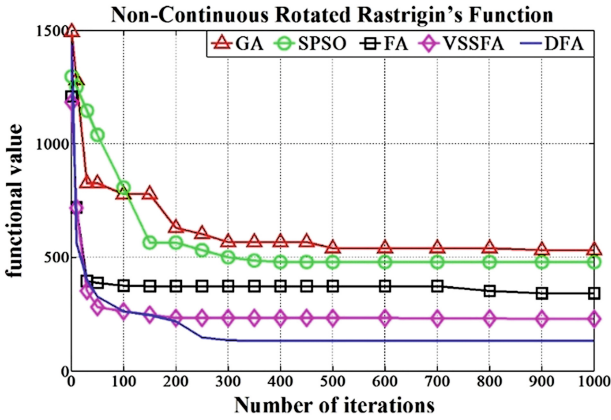


Fig. 4. Convergence curve of f_{13}

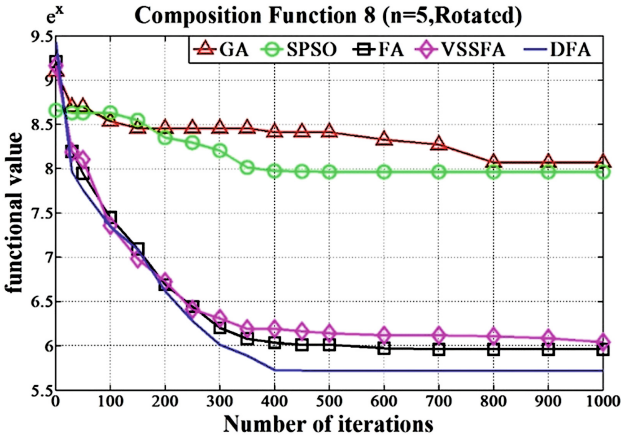


Fig. 5. Convergence curve of f_{28}

5 Conclusion

In the paper, to address the problem that SFA easy to fall into local optimum, an improved firefly algorithm, named as DFA, was proposed. The proposed algorithm based on detecting firefly expanded the searching range of the firefly swarm, so that it is easier to find the global optimal value. Further, that the optimal location was added to the movement strategy highlighted the role of optimal location in the algorithm. Finally, the experiments on 28 benchmark functions show that DFA has a significant improvement on standard FA.

Acknowledgments. This paper is supported by the National Natural Science Foundation of China (61502290, 61401263), Industrial Research Project of Science and Technology in Shaanxi Province(2015GY016), the Fundamental Research Funds for the Central Universities, Shaanxi Normal University (GK201501008) and China Postdoctoral Science Foundation (2015M582606).

References

1. Yang, X.S.: Firefly algorithm. Nature-Inspired Metaheuristic Algorithms Second Edition. Luniver Press, Bristol (2010)
2. Cheung, N.J., Ding, X.M., Shen, H.B.: Adaptive firefly algorithm: parameter analysis and its application. *PLoS ONE* **9**(11), e112634 (2014)
3. Yang, X.S.: Firefly algorithm, stochastic test functions and design optimization. *Int. J. Bio-Inspired Comput.* **2**(2), 78–84 (2010)
4. Johari, N.F., Zain, A.M., Mustaffa, N.H.: Firefly algorithm for optimization problem. *Appl. Mech. Mater.* **421**, 512–517 (2013)
5. Yang, X.S.: Firefly algorithms for multimodal optimization. *Stochast. Algorithms Found. Appl.* **5792**, 169–178 (2010)
6. Maheshwar Kaushik, K., Arora, V.: A hybrid data clustering using firefly algorithm based improved genetic algorithm. *Procedia Comput. Sci.* **58**, 249–256 (2015)
7. Farook, S.: Regulating LFC regulations in a deregulated power system using hybrid genetic-firefly algorithm. In: *IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, IEEE International Conference, Coimbatore, India, pp. 1–7 (2015)
8. Arora, S., Singh, S., Singh, S., Sharma, B.: Mutated firefly algorithm. In: *International Conference on Parallel, Distributed and Grid Computing*, pp. 33–38 (2014)
9. Yu, S., Zhu, S., Ma, Y., Mao, D.: A variable step size firefly algorithm for numerical optimization. *Appl. Math. Comput.* **263**, 214–220 (2015)
10. Tilahun, S.L., Ong, H.C.: Modified firefly algorithm. *J. Appl. Math.* **467631**(12), 2428–2439 (2012)
11. Tian, Y., Gao, W., Yan, S.: An improved inertia weight firefly optimization algorithm and application. *Int. Conf. Control Eng. Commun. Technol. Liaoning China* **4**, 64–68 (2012)
12. Bidar, M., Kanan, H.R.: Jumper firefly algorithm. In: *3rd International Conference on Computer and Knowledge Engineering (ICCKE)*, Mashhad, Iran, pp. 267–271 (2013)
13. Fateen, S.E.K.: Intelligent firefly algorithm for global optimization. In: Yang, X.-S. (ed.) *Cuckoo Search and Firefly Algorithm*. *SCI*, vol. 516, pp. 315–330. Springer, Heidelberg (2014)
14. Tuba, M., Bacanin, N.: Upgraded firefly algorithm for portfolio optimization problem. In: *UK Sim-AMSS 16th International Conference on Computer Modelling and Simulation*, Cambridge, USA, pp. 113–118 (2014)
15. Yang, X.S.: Firefly algorithm, Levy flights and global optimization. In: Bramer, M., Ellis, R., Petridis, M. (eds.) *Research and Development in Intelligent Systems XXVI*, Springer London, pp. 209–218 (2010)
16. Gandomi, A.H., Yang, X.S., Talatahari, S., Alaiv, A.-H.: Firefly algorithm with chaos. *Commun. Nonlinear Sci. Numer. Simul.* **18**(1), 89–98 (2013)
17. Baykasoglu, A., Ozsoydan, F.B.: Adaptive firefly algorithm with chaos for mechanical design optimization problems. *J Appl. Soft Comput.* **36**(c), 152–164 (2015)

18. Liu, C.N., Tian, Y.F., Zhang Q., Yuan J., Xue, B.B.: Adaptive firefly optimization algorithm based on stochastic inertia weight. In: Sixth International Symposium on Computational Intelligence and Design, Hangzhou, China, pp. 334–337 (2013)
19. Zhang, Y.N., Teng, H.F.: Detecting particle swarm optimization. *Concurrency Comput. Pract. Experience* **21**(4), 449–473 (2009)
20. Liang, J.J., Qu, B.Y., Suganthan, P.N., Hernández-Díaz, A.G.: Problem Definitions and Evaluation Criteria for the CEC 2013 Special Session on Real-Parameter Optimization